CAF Station performance report.

Understand the resource usage by basic components of the SAM data handling system in CAF environment. Identify bottlenecks to improve data throughput.

Basic data

Files processed.

```
[sam@cdfsam09 station__cdfsam09__station_test_int__cdf-int]$ awk 'BEGIN { sum = 0 }/closes/{ sum = sum + 1; } END { print sum}' trace
```

39347

Test time.

33440 seconds

Absolute times

Time Station spent in DB per file

```
[sam@cdfsam09 station__cdfsam09__station_test_int__cdf-int]$ awk '/SamClock/' trace | grep -v Status |  awk 'BEGIN { sum = 0 } {  sum=sum+strtonum($NF); }END { print sum,NR}'
16075.5 155766
```

16075.5/39347 = **0.40855719622842912**

Time DB spent per file

```
[sam@cdfsam09 station__cdfsam09__station_test_int__cdf-int]$ awk '/SamClock/' trace |  awk 'BEGIN { sum = 0 } {  sum=sum+strtonum($NF); }END { print sum,NR}'
```

24300.8 234460

per file:
24300.8/39347 = **0.61760235850255418**

Total time spent on a file.

33440/39347 = **0.85** seconds.

Relative CPU times.

Station CPU time per file

Total CPU time = 1h 10min 1262.724 Mhz
(1*3600+10*60)/39347. = **0.10674257249599715**

DB server CPU time per file

Total CPU time = 2H 30min, 1262.709 Mhz
CPU time spent per file: (2*3600+30*60)/39347. = **0.22873408391 99938**

Avg number of used Oracle connections

```
awk 'BEGIN { sum=0 }/nAvailableDbConnections/{sum = sum + strtonum($NF); }
END { print sum/NR}' dbg-SAMDbServer.station_test_int.05_08_12-00_00_00.3912-2
3.6266323751028033
```

5 - 3.6266323751028033 = **1.3733676248971967**

Analysis

Absolute Oracle time per file:

<DB time per file> - <DB server overhead> = <time spent in oracle DB>
0.61 - 0.22 = **0.39**

Impact of running processes on Oracle connection availability.

Considering the fact that there were 1000 consumers and 20 projects why connection usage was so low ?
Most of the time projects do not update DB in parallel due to the station limit in the file delivery rate.

DB time and station time do not add up to 0.85. (0.1 + 0.4 != 0.85)

Where did the rest of the time go?
Possible answers:

1. The absolute time spent by station in the DB was 0.4. The relative time spent by station on a file was 0.1. As 0.1 + 0.4 != 0.85 it follows that we may need to factor in the machine load to convert relative station time into machine absolute time. In subsequent tests I increased polling interval virtually eliminating any load

on a machine. Despite that the timing did not improve and stayed on the level of 0.85 sec per file.

2. Station wasted time waiting. I was probing this hypothesis by investigating station state in run time as well as doing thorough inspection of the code that was responsible for internal message loop scheduling. In the end I have redesigned the scheduler of the CORBA event loop to eliminate inefficiencies and unnecessary micro waits that I found on the course.

Following 2, the station version (**v6_0_2_3**) was released and installed at cdfsam09 for testing.

While testing, station showed much improved absolute file processing time that measured 0.54 seconds per file. The new time appears 50% improvement over the previous result and is consistent with the "down to top" analysis.

Conclusions.

Inefficiencies in the station CORBA event scheduling algorithm multiplied station time spent in the database by factor of 2. Before redesign, the absolute file processing time was close to 2*<DB time per file> + <station CPU time per file>. After the scheduler redesign the absolute file processing time became close to <DB time> + <station CPU time per file>. Thus the change reduced station dependency on DB server.

The Average number of connections used was 1.37. This is clear underutilization of the available Oracle resources. Parallelization of the independent DB calls is one way to address this issue. Approach is covered in previous report.

The breakdown of times in the DB server: 0.22 (infrastructure), 0.39 (oracle). If the infrastructure improves Oracle connection usage the DB server time overhead will surpass the Oracle lag. Assuming connection utilization is over 2, DB server becomes the major limiting factor. At that point it will make sense to distribute the load among multiple DB servers (DB server multiplexing) to better utilize Oracle connection pool.